

Lab 074 : Foundry Agent Service ? Simulation d'exploitation en production

Niveau : L300 Parcours : ? Microsoft Foundry Durée : ~120 min ? Coût : Gratuit ? Utilise des données d'agents simulées

Ce que vous apprendrez

Ce qu'est le Foundry Agent Service et comment il orchestre les systèmes multi-agents en production Comment les types d'agents (spécialiste, orchestrateur) collaborent dans une architecture de production Analyser la santé de la flotte d'agents : volumes de requêtes, latence, taux d'erreur et statut Identifier les agents dégradés et les risques de configuration (par ex. filtres de contenu désactivés) Construire un tableau de bord de santé de la flotte à partir de données de monitoring simulées

Introduction

Le Azure AI Foundry Agent Service fournit une plateforme managée pour déployer, orchestrer et surveiller des systèmes multi-agents à l'échelle de l'entreprise. Au lieu de construire une orchestration personnalisée, vous définissez des agents avec des outils, une mémoire et des modèles spécifiques ? et le service gère le routage, la gestion d'état et la mise à l'échelle.

Types d'agents

Type | Rôle | Exemple | Orchestrateur | Route les requêtes vers les spécialistes, gère le flux de conversation | SupportRouter, Coordinator | Spécialiste | Gère un domaine spécifique avec des outils et une mémoire dédiés | ProductAdvisor, OrderProcessor |

Le scénario

Vous êtes un SRE plateforme gérant un déploiement multi-agents pour une entreprise de commerce en ligne. La flotte comprend 8 agents ? 2 orchestrateurs et 6 spécialistes ? fonctionnant sur Azure Container Apps. Vous avez été alerté qu'un agent est dégradé et devez enquêter.

Votre jeu de données (foundry_agents.csv) contient le statut actuel de la flotte. Votre mission : analyser les métriques de santé, identifier les problèmes et produire un rapport de statut de la flotte.

info "Données simulées" Il s'agit d'un lab d'analyse opérationnelle, pas d'un lab de déploiement réel. Il utilise un CSV de flotte d'agents simulé qui reflète les métriques que vous verriez dans le tableau de bord de monitoring d'Azure AI Foundry. Les patterns (pics de latence, taux d'erreur, statut dégradé) représentent des scénarios courants en production.

Prérequis

Exigence | Raison | Python 3.10+ | Exécuter les scripts d'analyse | Bibliothèque pandas | Manipulation des données |

[diagram or code sample]

tip "Démarrage rapide avec GitHub Codespaces" Open in GitHub Codespaces

Toutes les dépendances sont pré-installées dans le devcontainer.

? Fichiers de support

note "Téléchargez ces fichiers avant de commencer le lab" Enregistrez tous les fichiers dans un dossier lab-074/ de votre répertoire de travail.

Fichier | Description | Téléchargement | broken_foundry.py | Exercice de correction de bugs (3 bugs + auto-tests) | ? Télécharger | foundry_agents.csv | Jeu de données | ? Télécharger |

Étape 1 : Comprendre l'architecture de la flotte

Avant d'analyser les données, comprenez comment les agents s'articulent :

[diagram or code sample]

Champs de configuration clés

Champ | Description | memory_type | Comment l'agent persiste son état : cosmos_db (durable), ai_search (vectoriel), session_only (éphémère), none | deployment | Infrastructure : container_apps (managé) ou vm (auto-hébergé) | content_filter | Si la sécurité du contenu Azure AI est enabled ou disabled | status | Santé de l'agent : active ou degraded |

Étape 2 : Charger et explorer les données de la flotte

[diagram or code sample]

Sortie attendue :

[diagram or code sample]

Étape 3 : Analyser le volume de requêtes et la distribution de charge

Comment le trafic est-il distribué dans la flotte ?

[diagram or code sample]

Sortie attendue :

[diagram or code sample]

Agent | Requetes | Part | Coordinator | 3 200 | 34,1 % | SupportRouter | 2 100 | 22,4 % | ProductAdvisor | 1 250 | 13,3 % | OrderProcessor | 890 | 9,5 % | QualityReviewer | 780 | 8,3 % | InventoryMonitor | 560 | 6,0 % | AnalyticsAgent | 420 | 4,5 % | LegacyBridge | 180 | 1,9 % |

tip "Insight" Le Coordinator orchestrateur gère 34 % de tout le trafic ? c'est le point d'entrée de la plupart des requêtes. S'il tombe en panne, tout le système est affecté. Le SupportRouter est le deuxième plus sollicité, routant les requêtes de support client vers les spécialistes.

Étape 4 : Identifier les agents dégradés et à risque

4a ? Agents dégradés

[diagram or code sample]

Sortie attendue :

[diagram or code sample]

4b ? Agents avec taux d'erreur élevé

[diagram or code sample]

4c ? Statut du filtre de contenu

[diagram or code sample]

warning "Risque de sécurité" LegacyBridge (FA08) a son filtre de contenu désactivé et fonctionne sur une VM auto-hébergée. C'est un risque de conformité ? tous les agents en production devraient avoir la sécurité du contenu activée, en particulier ceux qui traitent des données clients.

Étape 5 : Analyser les patterns de mémoire et d'infrastructure

[diagram or code sample]

[diagram or code sample]

Étape 6 : Construire le rapport de santé de la flotte

[diagram or code sample]

? Exercice de correction de bugs

Le fichier lab-074/broken_foundry.py contient 3 bugs qui produisent des métriques de flotte incorrectes. Pouvez-vous les trouver et les corriger tous ?

Exécutez les auto-tests pour voir lesquels échouent :

[diagram or code sample]

Vous devriez voir 3 tests échoués. Chaque test correspond à un bug :

Test | Ce qu'il vérifie | Indice | Test 1 | Total des requêtes sur 24h | Devrait sommer les

requêtes, pas les moyenner | Test 2 | Nombre d'agents dégradés | Devrait compter le statut degraded, pas active | Test 3 | Agents sans mémoire durable | Devrait compter none/session_only, pas cosmos_db |

Corrigez les 3 bugs, puis relancez. Quand vous voyez All passed!, c'est terminé !

? Vérification des connaissances

question "Q1 (Choix multiple) : Quel est le rôle d'un agent orchestrateur dans un déploiement multi-agents Foundry ?"

- A) Il effectue une tâche de domaine spécifique comme le traitement des commandes - B) Il route les requêtes vers les agents spécialistes et gère le flux de conversation - C) Il stocke la mémoire de l'agent dans Cosmos DB - D) Il surveille la santé des agents et redémarre les agents en échec

??? success "? Révéler la réponse" Correct : B) Il route les requêtes vers les agents spécialistes et gère le flux de conversation

Les agents orchestrateurs agissent comme le « contrôleur de trafic » dans un système multi-agents. Ils reçoivent les requêtes entrantes, déterminent quel(s) spécialiste(s) doivent les traiter, routent la conversation en conséquence et gèrent le flux global. Les spécialistes gèrent le travail spécifique au domaine.

question "Q2 (Choix multiple) : Pourquoi un filtre de contenu désactivé est-il un risque de sécurité pour les agents en production ?"

- A) Cela rend l'agent plus lent - B) Cela permet à l'agent de générer du contenu nuisible, biaisé ou violant les politiques - C) Cela empêche l'agent d'accéder aux API externes - D) Cela augmente les coûts en tokens

??? success "? Révéler la réponse" Correct : B) Cela permet à l'agent de générer du contenu nuisible, biaisé ou violant les politiques

Les filtres Azure AI Content Safety détectent et bloquent le contenu nuisible (discours haineux, violence, automutilation, contenu sexuel). Désactiver le filtre signifie que l'agent peut produire ou répondre à un tel contenu sans garde-fous ? un risque de conformité et de réputation dans tout déploiement de production.

question "Q3 (Exécutez le lab) : Quel est le nombre total de requêtes dans toute la flotte au cours des dernières 24 heures ?"

Exécutez l'analyse de l'étape 3 sur ? foundry_agents.csv et vérifiez les résultats.

??? success "? Révéler la réponse" 9 380 requêtes

Somme de toutes les valeurs requests_24h des agents : $1\,250 + 890 + 2\,100 + 560 + 3\,200 + 780 + 420 + 180 = 9\,380$.

question "Q4 (Exécutez le lab) : Combien d'agents sont dans un état dégradé ?"

Exécutez l'analyse de l'étape 4a pour le découvrir.

??? success "? Révéler la réponse" 1 agent

Seul AnalyticsAgent (FA07) est dans un état degraded, avec un taux d'erreur de 8,5 % et une latence moyenne de 850 ms ? nettement pire que les autres agents. Cela indique probablement un problème de connectivité backend avec son magasin de mémoire AI Search.

question "Q5 (Exécutez le lab) : Combien d'agents ont leur filtre de contenu désactivé ?"

Exécutez l'analyse de l'étape 4c pour vérifier le statut du filtre de contenu.

??? success "? Révéler la réponse" 1 agent

Seul LegacyBridge (FA08) a content_filter=disabled. C'est aussi le seul agent déployé sur une VM auto-hébergée plutôt que sur Container Apps, et il a le taux d'erreur le plus élevé (12,0 %) de la flotte. Cet agent nécessite une attention immédiate.

Résumé

Sujet | Ce que vous avez appris | Foundry Agent Service | Plateforme managée pour

l'orchestration et le déploiement multi-agents | Types d'agents | Les orchestrateurs routent ; les spécialistes exécutent les tâches de domaine | Monitoring de la flotte | Suivre les requêtes, la latence, les taux d'erreur et le statut par agent | Détection de dégradation | Identifier les agents avec des taux d'erreur ou une latence élevés | Sécurité du contenu | Tous les agents en production devraient avoir les filtres de contenu activés | Patterns de mémoire | Cosmos DB pour le durable, AI Search pour le vectoriel, session_only pour l'éphémère |

Prochaines étapes

Lab 034 ? Multi-agents avec Semantic Kernel (construire les agents eux-mêmes) Lab 033 ?

Observabilité des agents avec Application Insights (monitoring approfondi) Lab 030 ?

Foundry Agent + MCP (connecter les agents à des outils externes) Lab 075 ? Power BI Copilot (visualiser les données de flotte avec des tableaux de bord assistés par IA)

Modernisation de Phase 3

warning "Verified 2026-10-27" Utilisez Microsoft Foundry et Agent Service GA fondé sur Responses API. Les évaluations sont GA ; les hosted agents restent en preview et doivent être signalés. Ne dépendez pas des Workflows du portail retirés le 01/12/2026.

[diagram or code sample]

Foundry Agent Service GA announcement Foundry Agent Service Foundry evaluation Foundry Workflows retirement

Phase 3 · Simulation / Parcours Réel

Simulation Parcours Réel

Preserve the existing free simulation and its fixtures. Estimated cost: US\$0.

Reuse the same acceptance criteria against the documented hosted service. Before running, record a service-specific estimate and a hard spending cap. After validation: Supprimez déploiements, points de terminaison, index, traces et données de test après validation.

Kit d'accompagnement

Le build du site génère des téléchargements PDF et ZIP stables. Le ZIP contient le guide du lab et tous les fichiers d'accompagnement déclarés.

Télécharger le PDF généré Télécharger le ZIP généré

Supporting file: broken_foundry.py

```
/usr/bin/env python3 """Bug-fix: Broken Foundry Analyzer - Lab 074. Fix 3 bugs.""" import
io, pandas as pd def total_requests(agents): # Bug #1: Returns average instead of sum
return int(agents["requests_24h"].mean()) def count_degraded(agents): # Bug #2: Counts
'active' instead of 'degraded' return (agents["status"] == "active").sum() def
agents_without_memory(agents): # Bug #3: Counts cosmos_db (has memory) instead of
none/session_only return (agents["memory_type"] == "cosmos_db").sum() def run_tests():
csv="agent_id,status,requests_24h,memory_type\nF1,active,1000,cosmos_db\nF2,degraded,500,ses
sion_only\nF3,active,800,none\nF4,active,1200,cosmos_db"
a=pd.read_csv(io.StringIO(csv));p,f=0,0 t=total_requests(a);exp=3500 if t==exp:print(f"OK
T1: {t}");p+=1 else:print(f"FAIL T1: {t} (exp {exp})");f+=1 d=count_degraded(a);exp=1 if
d==exp:print(f"OK T2: {d}");p+=1 else:print(f"FAIL T2: {d} (exp {exp})");f+=1
n=agents_without_memory(a);exp=2 if n==exp:print(f"OK T3: {n}");p+=1 else:print(f"FAIL T3:
{n} (exp {exp})");f+=1 print(f"\n{'All passed!' if f==0 else f'failed'}") if
__name__=="__main__": print("Tests:\n");run_tests()
```

Supporting file: foundry_agents.csv

agent_id,agent_name,agent_type,model,tools_count,memory_type,deployment,status,requests_24h,

avg_latency_ms,error_rate_pct,content_filter

FA01,ProductAdvisor,specialist,gpt-4o,5,cosmos_db,container_apps,active,1250,320,1.2,enabled
FA02,OrderProcessor,specialist,gpt-4o,8,cosmos_db,container_apps,active,890,450,2.1,enabled
FA03,SupportRouter,orchestrator,gpt-4o-mini,3,session_only,container_apps,active,2100,180,0.
5,enabled FA04,InventoryMonitor,specialist,gpt-4o-mini,4,ai_search,container_apps,active,560
,220,0.8,enabled FA05,Coordinator,orchestrator,gpt-4o,2,cosmos_db,container_apps,active,3200
,280,1.0,enabled FA06,QualityReviewer,specialist,gpt-4o,3,session_only,container_apps,active
,780,350,1.5,enabled FA07,AnalyticsAgent,specialist,gpt-4o,6,ai_search,container_apps,degrad
ed,420,850,8.5,enabled FA08,LegacyBridge,specialist,gpt-4o-mini,10,none,vm,active,180,1200,1
2.0,disabled